

A0521/7189  
PJG/aml  
442475.1

A patent application for:

**AUTHORING SYSTEM FOR COMBINING TEMPORAL AND  
NONTEMPORAL DIGITAL MEDIA**

by

Peter A. Balkus  
130 Nonset Path  
Acton, Massachusetts 01720  
Citizen of the United States of America

Glenn McElhoe  
1 Carl Road  
Arlington, Massachusetts 02474  
Citizen of the United States of America

T. Winton Crofton  
37 Winchester Road  
Newton, Massachusetts 02458  
Citizen of the United States of America

and

Thomas C. Purcell  
324 Jenness Pond Road  
Northwood, New Hampshire 03261  
Citizen of the United States of America

**EXPRESS MAIL NO.** EL506834911US  
(03/31/00)

007330 64265500

## AUTHORING SYSTEM FOR COMBINING TEMPORAL AND NONTEMPORAL DIGITAL MEDIA

### BACKGROUND

5 A variety of systems are used for authoring multimedia presentations such as motion pictures, television shows, advertisements for television, presentations on digital versatile disks (DVDs), interactive hypermedia, and other presentations. Such authoring systems generally provide a user interface and a process through which multimedia data is captured and stored, and through which the multimedia presentation is created, reviewed and published  
10 for distribution. The user interface and process for authoring generally depend on the kind of presentation being created and what the system developer believes is intuitive and enables an author to work creatively, flexibly and quickly.

Some multimedia presentations are primarily nontemporal presentations. That is, any change in the presentation typically depends on user activity or other event, instead of the  
15 passage of time. Some nontemporal multimedia presentations may include temporal components. For example, a user may cause a video to be displayed that is related to a text document by selecting a hyperlink to the video in the document.

Other multimedia presentations are primarily temporal presentations incorporating audio and/or video material, and optionally other media related to the temporal media.  
20 Primarily temporal media presentations that are well known today include streaming media formats such as QuickTime, Real Media, Windows Media Technology and SMIL, and formats that encode data in the vertical blanking interval of a television signal, such as used by WebTV, ATVEF, and other similar formats.

A variety of authoring tools have been developed for different kinds of presentations.  
25 Tools for processing combined temporal and nontemporal media include those described in PCT Publication No. WO99/52045, corresponding to U.S. Patent Application Serial No. 09/054,861, and PCT Publication No. WO96/31829, corresponding to U.S. Patent Application Serial No. 08/417,974, and U.S. Patent 5,659,793 and U.S. Patent 5,428,731.

### SUMMARY

30 An authoring tool has a graphical user interface enabling interactive authoring of a multimedia presentation including temporal and nontemporal media. The graphical user interface enables specification of the temporal and spatial relationships among the media and

00539749 033100  
00T EEO 6426E560



It is also possible to associate a streaming media presentation with another streaming media presentation. For example, an event in one streaming media presentation may be used to initiate playback of another subsequent streaming media presentation. The two presentations may have different layout specifications. A document in a markup language may be created to include a hyperlink to each of the plurality of streaming media presentations.

### BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is an illustration of an example multimedia presentation;

Fig. 2 is an illustration of a relationship among multiple presentations;

Fig. 3 is an illustration of a timeline for defining a multimedia presentation;

Fig. 4 illustrates example layouts for a multimedia presentation;

Fig. 5 is an illustration of an example graphical user interface for specifying a layout;

Fig. 6 is an illustration of an example graphical user interface for specifying a

mapping between frames in a layout and tracks in a timeline;

Fig. 7A is a data flow diagram illustrating a relationship of parts of a system for authoring and publishing a multimedia presentation;

Fig. 7B is an illustration of an example graphical user interface for interactively authoring and viewing a presentation;

Fig. 8A illustrates an architecture for implementing an editing viewer of Fig. 7A;

Fig. 8B illustrates an architecture for implementing a display manager of Fig. 8A;

Fig. 8C is a flowchart describing how a graphical user interface may be constructed;

Fig. 8D is a flowchart describing how a display manager may display contents and its corresponding portion of the editing interface;

Fig. 8E is a flowchart describing how the table of contents display may be updated;

Fig. 8F is a flowchart describing how a new table of contents file may be generated;

Fig. 9 is a flowchart describing how a presentation may be published;

Fig. 10 illustrates a graphical user interface for managing a transfer process of a multimedia presentation;

Fig. 11A is a flowchart describing how a presentation may be encoded;

Fig. 11B is a flowchart describing, in one implementation, how a program may be encoded;

Fig. 11C is a flowchart describing how a presentation may be transferred;

Fig. 12 is a data flow diagram illustrating interaction of a transfer tool with a streaming server and a data server; and

Fig. 13 is a data flow diagram illustrating a relationship of multiple editing and transfer systems with a streaming server.

#### DETAILED DESCRIPTION

In this description, all patent applications and published patent documents referred to herein are hereby incorporated by reference.

Referring to Fig. 1, an example of a multimedia presentation, which may be created using an authoring system to be described herein, will now be described. In general, a multimedia presentation is a combination of temporal media, such as video, audio and computer-generated animation, and nontemporal media, such as still images, text, hypertext documents, etc. Some temporal media, such as animations in the GIF format or the Macromedia Flash formats may be used as if they were nontemporal media. The temporal and nontemporal media may be combined in many different ways. For example, a multimedia presentation may include audio and/or video combined with multimedia slides that are time synchronized with the audio and/or video. The presentation also may include advertisements and/or an index of the temporal media. In general, there is a temporal relationship and a spatial relationship among the temporal and nontemporal media. In some presentations, only a temporal relationship exists between certain temporal media, such as audio, and the nontemporal media. An example presentation shown in Fig. 1, includes video 100, HTML events 102, a table of contents 104, and an advertisement 106.

Fig. 2 illustrates a more complex multimedia presentation format. This multimedia presentation includes a hypermedia document 200, for example in a markup language, including hyperlinks to one or more streaming media presentations, as indicated at 202, 204, and 206. Upon selection of a hyperlink, the corresponding streaming multimedia presentation 208, 210 or 212 may be played. An event at or near the end of a streaming multimedia presentation may be used to initiate playback of the subsequent multimedia presentation. The different presentations may have different specified spatial relationships.

There are many ways in which such multimedia presentations may be stored. For example, various streaming media formats, such as Real Media, Microsoft Windows Media

Technology, QuickTime and SMIL, may be used. The temporal media also may be encoded in a television signal, with nontemporal media encoded in a vertical-blanking interval of the television signal, such as used by WebTV, ATVEF and other formats.

Creating such a multimedia presentation involves creating a temporal relationship between each element of nontemporal media and the temporal media. Such a relationship may be visualized using a timeline, an example of which is shown in Fig. 3. In general, a timeline has one or more tracks of temporal media, and one or more tracks of nontemporal media. For example, there may be one video track, one audio track, and an event track. The presentation of the media on all the tracks is synchronized by the positions of the elements in the timeline. These positions may be specified graphically through a graphical user interface. Various data structures may be used to represent such a timeline, such as those described in U.S. Patent 5,584,006 (Reber), U.S. Patent 5,724,605 (Wissner) and PCT Publication No. WO98/05034.

The timeline is a time based representation of a composition. The horizontal dimension represents time, and the vertical dimension represents the tracks of the composition. Each track has a row in the timeline which it occupies. The size of a displayed element in a graphical user interface is determined as a function of the duration of the segment it represents and a timeline scale. Each element in each track of the timeline has a position (determined by its start time within the presentation), a title and associated data and optionally a duration.

Fig. 3 illustrates an example timeline which includes two audio tracks 300, two video tracks 302, two event tracks 304, a title track 306, and a table of contents track 308. Each of these tracks will now be described.

An audio track 300 or a video track 302 is for placement of temporal media. Such tracks commonly are used in video editing applications, such as shown in PCT Publication No. WO98/05034, which corresponds to U.S. Patent Application Serial Nos. 08/687,926 and 08/691,985. Similarly, a title track 306 commonly is used to create title effects for movies, such as scrolling credits. As such, titles commonly are considered temporal media because they have parameters that are animated over time and that are combined with video data. Each track supports defining a sequence of segments of media data. A segment references, either directly or indirectly, the media data for the segment.

In the timeline shown herein, event tracks 304 associate nontemporal media with a particular point in time, thus creating a temporal relationship with the temporal media in tracks 300, 302, and 306. Each event track is a list of events. Each event includes a time and references a data file or a uniform resource locator, either directly or indirectly, from which media data for the event may be received.

The table of contents track 308 associates a table of contents entry with a point in time. The table of contents may be used as an index to the temporal media. Each entry includes a time and associated content, typically text, entered by the author. As described in more detail below, the table of contents entries are combined into a single document for display. If a user selects an element in the table of contents as displayed, the presentation is displayed starting at the point in time corresponding to the selected element.

The spatial relationship of the elements in the timeline as presented also may be specified by the author. In one simple example, a layout specification indicates a combination of frames of a display area, of which one or more frames is associated to one or more of the tracks in the timeline. Some tracks might not be associated with a display frame. Some frames might be associated directly with static media and not with a track. In general a frame is associated with only one track and a track is associated with only one frame.

The possible combinations and arrangements of the various tracks in a timeline are unlimited, and are not limited to visual media. As shown in the examples in Fig. 4, the visual display may be merely a table of contents 400, or an event track 402, or both 404, for example, in combination with audio. These examples are merely illustrative. In some cases, the audio has a corresponding visual component that may be displayed, such as volume and position controls. Video may be displayed, for example, with an event track 406, or a table of contents track 408, or both 410, such as shown in Fig. 4.

A graphical user interface, and example of which is described in connection with Fig. 5, enables a user to select from among several layout specifications that have been stored as templates. A graphical user interface, an example of which is described in connection with Fig. 6, enables an author to make assignments between tracks in the timeline and frames in the display.

In Fig. 5, a graphical user interface 500 illustrates templates in a template window 502. A template defines a mapping between frames and tracks and a display arrangement of the frames such as described in Fig. 4. A selected template such as 504 is viewed in a

preview pane 506. A user may browse the file system to identify other templates by selecting a button 508 as in conventional user interfaces. A template may be defined using the hypertext markup language (HTML), for example by using frame set definitions. A template may be authored using any conventional HTML authoring tool, word processor or text editor.

5 In the user interface, a template file may be accessed to determine its frame set definitions to generate an appropriate icon for display. Similarly, the preview pane 506 is generated by accessing the frame set definition within the selected template file. The mapping between frames and tracks also is stored in the template file.

An example template file follows:

```
10 <HTML>
    <AVIDPUB tagtype="framemap" frameName="Frame_A" feature="MOVIE"
    originalurl="static.htm">
    <AVIDPUB tagtype="framemap" frameName="Frame_B" feature="EVENTTRACK"
15 featurenum="1" >
    <AVIDPUB tagtype="framemap" frameName="Frame_C" feature="EVENTTRACK"
    featurenum="2" >
    <AVIDPUB tagtype="framemap" frameName="Frame_D" feature="TOC"
    originalurl="static.htm">
20 <AVIDPUB tagtype="framemap" frameName="Frame_E" feature="EVENTTRACK"
    featurenum="3" >
    <AVIDPUB tagtype="framemap" frameName="Frame_Top" feature="STATICHTML"
    featurenum="0" originalurl="header.htm">
    <FRAMESET cols="40%,60%" bordercolor="blue" frameborder=yes framespacing=2>
25     <FRAMESET rows="70,40%,*">
        <FRAME SRC="header.htm" name="Frame_Top">
        <FRAME SRC="AvidVid.htm" name="Frame_A">
        <FRAME SRC="AvidPubToc.html" name="Frame_D">
    </FRAMESET>
30 <FRAMESET rows="33%,34%,*">
        <FRAME SRC="static.htm" name="Frame_B">
        <FRAME SRC="static.htm" name="Frame_C">
        <FRAME SRC="static.htm" name="Frame_E">
    </FRAMESET>
35 </FRAMESET>
</HTML>
```





editing. The editing manager, given a point in time 722 on the timeline and optionally a playback rate 724 from the editing GUI 700, generates video data 714 and other visible data 710 for display in the editing GUI 700, in an arrangement defined by the layout specification 706, using media files 712. An example implementation of the editing manager is described below in connection Figs. 8A-F. After the author has completed creating the presentation, the publisher 718 is invoked to process the timeline 716, layout specification 706, and media file 712 to generate the published presentation 720.

An example GUI for the editing GUI of Fig. 7A will now be described in connection with Fig. 7B. In Fig. 7B, the timeline region 700 includes an index track 702, a video track 704, a titles track 706, two audio tracks 708 and 710, three event tracks 712, 714 and 716 and the timeline scale 718. The timeline scale determines the number of pixels that represents a time unit. Increasing or decreasing this time scale allows the user to focus on a particular location in the composition, or to have a more of an overview of the composition. A viewer window 720 displays the video data and other visual information. A display controller 722 includes a position indicator 724 which points to the present position within the multimedia presentation which is being viewed. Forward and backward skip buttons 726 and 728 and play buttons 730 also may be provided. The position indicator 724 is associated with a position indicator 736 in the timeline 700. The buttons 726, 728 and 730, and position indicator 724 may be used to control the viewing of the multimedia presentation during authoring. Frame boundaries, as indicated at 732 and 734, to the frame set definitions in the layout specification. The frame boundaries 732 and 734 may be made adjustable using a cursor positioning device, such as a mouse or touchpad. Such adjustments may be transformed into edits of the layout specification. The various kinds of operations that may be performed to edit the audio and video and to add titles are described in more detail in PCT Publication No. WO98/05034.

How entries in the index or table of contents track 702 and event tracks 712 through 716 are added or modified will now be described. A region 740 illustrates available multimedia data for insertion into events. Buttons 742, 744 and 746 enable different views of the information presented in region 740. Button 742 selects a mode in which the system displays a picture of the data. Button 744 selects a mode in which the system displays a detailed list including a small picture, filename, and timestamp of the data file or resource. Button 746 selects a mode in which the system displays only titles. Other modes are possible

and the invention is not limited to these. The names displayed are for those files found in the currently active path in the file system used by the authoring tool or other resources available to the system. The list operation, for example, may involve a directory lookup performed by the computer on its file system. A user may select an indicated data file or resource and drag its icon to an event timeline either to create a new event, or to replace media in an existing event, or to add media to an existing event.

On the event timeline, an event 750 indicates a data file or other resource associated with a particular point in time. Event 752 indicates that no file or resource is associated with the event at this time. In response to a user selection of a point on an event track, a new event may be created, if one is not already there, or the selected event may be opened. Whether a new event is created, or an existing event is opened, the user may be presented with a properties dialog box to enable entry of information, such as a name for the event, or a file name or resource locator for the associated media, for storage into the event data structure. An event that is created may be empty, i.e., might not refer to any data file or resource.

The elements on the event track may be illustrated as having a width corresponding to the amount of time it would take to download the data file over a specified network connection. To achieve this kind of display, the number of bytes of a data file is divided by the byte-per-second rate of the network connection to determine a time value, in seconds, which is used to determine the width of the icon for the event to be displayed on the event track. Displaying the temporal width of an object provides information to the author about whether enough time is available at the location of distribution to download the data and to display the data at the desired time.

Similar to the events, a user may select an element on the table of contents track as indicated at 754. An item may be added by selecting a point on the table of contents track with a cursor control device. Upon selection, a dialog window is displayed through which the user may enter text for the selected element. Each of the elements in the table of contents track 702 is displayed in the frame 756 in the viewer 720.

To display the presentation to the author, for a given point in time of the presentation, the system determines which contents should be displayed. In the example shown in Fig. 7B, event 758 is currently being displayed from the event track in viewer frame 760. The video is being shown in frame 762. The table of contents elements are shown in frame 756. A viewer such as shown in Figs. 7A and 7B may be implemented in many ways, depending on the

availability of preexisting program components to be used, and the platform on which the viewer is implemented. An example implementation will now be described in connection Figs. 8A through 8E for use with a platform as specified below. In this implementation, the viewer uses an Internet Explorer browser component, available from Microsoft Corporation, to render the nontemporal media. Currently available browser components are capable of processing encoded streaming media files but not video and audio data defined using a timeline. Thus, the temporal media, in particular the audio and video, is rendered in a manner typical in video editing systems, such as described in PCT Publication No. WO98/05034. The viewer described herein reads a presentation and accesses data, audio and video files to produce an the presentation without an encoded streaming media file, thus simulating the operation of a browser that uses streaming media files.

Referring now to Fig. 8A, an architecture for this implementation is illustrated. This architecture includes an asset manager 8100 which manages access to data files 8102 used in the presentation. A clip manager 8104 maintains the timeline data structure 8106 in response to instructions from the user via the graphical user interface. Requests for access to information from the timeline 8106 by the presentation manager 8108 and display manager 8110 also are managed by the clip manager 8104. The presentation manager 8108 maintains the layout specification 8112 and other display files 8114. The other display files include files in a markup language that define the table of contents frame and the video frames. An example layout was described above in connection with Fig. 6. An example table of contents file and example video frame files, for the Real Media and Windows Media technology formats, are provided in Appendices I-III, the interrelationship of which will now be described.

There are several ways in which the table of contents may be constructed to allow actions on a table of contents entry to cause a change in the playback position in the video frame. One example is provided by the source code and Appendices I-III. In the table of contents page, a JAVA script function called "seekToEPMarker" takes either a marker number (for Windows Media technology) or a time in milliseconds (for Real Media) and calls a function "seekToVideoMarker" of its parent frame in its frame set. This function call actually calls the JAVA script function of the child frame of the table of contents' parent frame that includes the video player. That function receives the marker and the time in

milliseconds and generates the appropriate commands to the media player to initiate playback of the streaming media from the designated position.

Turning again to Fig. 8A, the display managers 8110 each are associated with a display window in the viewer and control displaying content in their respective windows. In general, the display managers access data from the presentation manager 8108 and clip manager 8104 to provide data to the graphical user interface 8116, in response to events that modify the timeline or the presentation of data in the timeline as received from the graphical user interface or the clip manager. The graphical user interface 8116 communicates with the clip manager, presentation manager and display manager to create and maintain the view of the timeline and the presentation in response to user inputs.

A display manager, in one implementation, is described in more detail in connection with Fig. 8B. The display manager includes a controller module 8200 which communicates with the graphical user interface, presentation manager and clip manager. To display a data file, the controller instructs a browser component 8202 to render data for display. The output of the browser component is processed by an image scaling module 8204 that scales the result to fit within the appropriate display region in the viewer.

Referring now to Fig. 8C, how the display of the presentation in the viewer may be created will now be described. In particular, the layout of the presentation is defined by the layout specification 8112. This layout specification is parsed 8300 to generate a tree-like representation of the layout. In particular, as shown in the example layout specification provided above, some frames are defined as subframes of other frame sets. This hierarchical definition of frames translates into a tree-like representation. For each nonleaf node in the tree, a splitter window is created 8302 in the presentation display region on the user interface. For each leaf node in the tree, a display window is created 8304 within its associated splitter window. This display window is instructed 8306 to display its content at time zero, i.e., the beginning, in the presentation to initialize the display. The display window has an associated display manager 8110.

How the display manager displays data given a specified time in the presentation will now be described in connection with Fig. 8D. In particular, the display manager receives 8400 a time T. For event tracks, the event that has most recently occurred in the presentation prior to time T is identified 8402. The data file for that event is then obtained 8404. The browser component is then instructed 8406 to render the received data file. The image

scaling module scales the image produced by the browser component, in 8408, which is then displayed 8410 in the associated window. For video information, this process involves identifying the sample from the data file for the segment that is in the presentation at the specified time. This sample is scaled and displayed. Because the table of contents file is not time dependent, it is simply rendered, scaled and displayed and step 8402 may be omitted.

After initialization, each display manager acts as a "listener" process that responds to messages from other components, such as the clip manager and graphical user interface, to update the display. One kind of update is generated if display controls in the graphical user interface are manipulated. For example, a user may modify the position bar on either the timeline or the viewer to initiate display from a different point in time T. In response to such a change, the graphical user interface or the clip manager may issue a message requesting the display managers to update the display given a different time T. Similarly, during editing, changes to the timeline data structure at a given point in time T cause the clip manager to instruct the display managers to update the display with the new presentation information at that point in time T.

Playback may be implemented using the same display mechanism. During either forward or reverse playback at a continuous or user-controlled rate, a stream of instructions to update the display at different points in time T may be sent to the display managers. Each display manager updates its region of the display at each of the specified times T which it receives from the clip manager or graphical user interface.

Although the table of contents generally is a single file without time dependency, during editing it may be modified, after which the display is updated. One implementation for modifying the table of contents display will now be described in connection with Figs. 8E and 8F. In Fig. 8E, a display manager for the table of contents receives 8500 a message from the clip manager that a table of contents entry has been added to the table of contents track. The display manager requests 8502 the presentation manager for a new table of contents file. After receiving 8504 the indication of the new table of contents file, the browser component is instructed 8506 to render the data file. The rendered data file is then scaled 8508 for display.

How the presentation manager generates a new table of contents file is described in Fig. 8F. The presentation manager receives 8600 a message requesting a new table of contents file. The presentation manager requests 8602 the table of contents track information



for the server, and the HTTP address or URL of the server from which a user will access the published presentation, also may be provided. The server information may be separate for both data files and streaming media files.

This encoding and transfer information may be stored by the transfer tool as a named profile for later retrieval for transferring other presentations. Such profile data may include, for example, the data defining settings for encoding, and the data defining settings for transfer of encoded data files.

When setting up each of the connections for transfer, the connection also may be tested to confirm its operation. This test process involves transferring a small file to the destination and confirming the ability of the system to read the file from the destination.

After setup, the presentation may be audited 901 to reduce the number of errors that may otherwise result during the encoding and/or transfer processes. Profile information, described below, the presentation, and other information may be reviewed for likely sources of errors. For example, titles and/or other effects may be checked to determine whether the title and/or effect has been rendered. The timeline data structure may be searched to identify the data files related to each event, segment, table of contents entry, etc., to determine if any file is missing. The events in the timeline may be compared to the video or audio or other temporal data track to determine if any events occur after the end of the video or audio or other temporal data track. The layout specification also may be compared to the timeline data structure to ensure that no events or other data have been defined on tracks that are not referred to in the layout specification. Results of these various tests on the layout and timeline data structures may be provided to the user. Information about the profile used for the transfer process also may be audited. For example, whether passwords might be used on the target server, and the other information about the accessibility of the target server may be checked. The target directory also may be checked to ensure that no files in the native file format of the authoring tool are present in the target directory. Various other tests may be performed in an audit process and the invention is not limited thereto.

After optional auditing, the presentation is encoded 902 by transforming the timeline data structures into a format used by a standard encoder, such as provided for the Real Media Player or Windows Media Technology. Such encoding is described in more detail below in connection with Figs. 11A and 11B. The encoded presentation optionally may be previewed 904. To support preview, during encoding the files used to encode, and that will ultimately



be transferred to each server, are collected locally. The presentation may be encoded first to support preview by referring to the local files. The files for the presentation then are transferred 906 to each server. Before transfer, if the presentation was encoded for local preview, the references to local files are translated into references to files on the destination servers. For example, the encoded streaming media file generally is provided to a streaming media server, whereas other data files referred to by the streaming media file are provided to a standard hypertext transfer protocol daemon (HTTPD) or web server. The transfer process is described in more detail below in connection with Fig. 11C. Finally, the transferred presentation may be previewed 908 from the remote site.

A graphical user interface for facilitating the publishing process described in Fig. 9 will now be described in connection with Fig. 10. A user may set profile data by selecting setup or options 1000. During set up, a profile may be recalled, created or edited, and the user may specify the file folder and server on which the presentation will be stored. In response to selection of the "do it" menu item, the screen shown in Fig. 10 is displayed. First the presentation and profile data are audited. After the auditing step is complete, a checkmark appears in an icon 1006. Next, encoding of the presentation may be started at 1008. A user may optionally select to preview the encoded presentation locally prior to transfer. By selecting button 1010, a preview of the presentation may be initiated. After preview, the icon 1012 includes a checkmark. During transfer, a user may select to overwrite files that have the same name on the destination server, as indicated at 1014. The user may initiate the transfer by selecting the button indicated at 1016. After completion, the icon 1018 includes a checkmark. Finally, after transfer, the user may view the presentation as transferred from the destination server by selecting button 1020.

Referring to Fig. 11A, encoding of a presentation will now be described. In general, most encoders have an application programming interface that generate an encoded file in response to commands to add samples of media to the presentation. The commands for adding samples generally include the type of media, the time in the presentation in which the media is to be added and the media data itself as inputs to the command. The sample for video data is usually a frame. The sample of audio data is usually several samples defining a fraction of a second. The data also may be, for example, a uniform resource locator (URL) or other data.

More particularly, an API has functions that: 1) enable opening the component, 2) optionally present the user with a dialog box interface to configure the component, 3) set settings of the component that control its behavior, 4) connect the component to a user visible progress bar and to the source of the data, 5) to initiate the component to start translating the data into the desired format, 6) write the desired format to a file, and 7) close the component if the process is complete. On the receiving side of the API, the system has code to respond to requests for data from the export or encode component. The export component generally accesses the time, track number, and file or URL specified by the user, which are obtained from the timeline data structure. To the extent that data interpretation or project-specific settings are used by the encoder, this information also may be made available through an API.

The video and audio may be encoded 1100 separately using standard techniques. The table of contents and event tracks are then processed. In particular, a list of event assets is generated 1102. An event asset is defined by its filename, track, and time in the presentation. The frame set is then accessed 1104 to obtain a list of tracks and frame names. The items in the event tracks are then added to the streaming media file using the filename for the event and the frame name for the event, at the indicated time for the event, in 1106. The filename for the event is its full path including either a full URL for remote files or an indicator of the disk volume for files that are accessed locally or over a local area network (LAN). In step 1106, the filenames and frame names inserted into the streaming media file are those in the destination to which the media file is being transferred. Therefore, the encoding is dependent in part on the transfer parameters. The list created in step 1102 may be sorted or unsorted.

Using Real Media, the table of contents track does not affect the streaming media file. Using Windows Media technology, however, marker codes are inserted for each table of contents entry, although no marker codes are inserted for events.

Referring to Fig. 11B, an implementation using the Real Media encoder will now be described. A Real Media encoder 1120 issues requests 1122 for samples at a specified time. In response to these requests, a presentation processor 1124 implements the process described in Fig. 11A, and returns a sample 1126 from an event that occurs in the presentation at a time closest to and after the requested time. The response 1126 also indicates a time at which the encoder 1120 should request the next sample. This time is the time corresponding to the sample which was returned by the presentation processor 1124. The list of event assets

created in 1102 in Fig. 11A may be sorted prior to initiating encoding with the real media encoder 1120, or may be sorted on the fly in response to requests 1122 from the Real Media encoder 1120. After the end of the presentation is reached, the encoded presentation 1128 is available.

5           The process of transferring data to the servers will now be described in connection with Fig. 11C. After setup and encoding have been completed, the transfer of the presentation starts with preparing 1130 lists of files or resources of the presentation. A first list includes the table of contents file, the video frame file and the index or template file and all of the files that these three files directly reference. A second list is all files destined for  
10   the streaming media server. A third list is all of the files and resources in events and all of the files and resources these events reference directly. Resources that are not directly available at the local machine may be omitted from the list. This third list uses the complete path name or URL for the file or resource. For the drives or servers used for the files in the third list, a base path is found 1132. New directories on the destination servers are then  
15   created 1134 using the base paths as subdirectories of the target directory on the server. Files is all three lists are then transferred 1136 to their respective destinations.

          A computer system with which the various elements of the system described above, either individually or in combination, may be implemented typically includes at least one main unit connected to both one or more output devices which store information, transmit  
20   information or display information to one or more users or machines and one or more input devices which receives input from one or more users or machines. The main unit may include one or more processors connected to a memory system via one or more interconnection mechanisms. Any input device and output device also are connected to the processor and memory system via the interconnection mechanism.

25           The computer system may be a general purpose computer system which is programmable using a computer programming language. Computer programming languages suitable for implementing such a system include procedural programming languages, object-oriented programming languages, combinations of the two, or other languages. The computer system may also be specially programmed, special purpose hardware, or an  
30   application specific integrated circuit (ASIC).

          In a general purpose computer system, the processor is typically a commercially available processor which executes a program called an operating system which controls the

00539749.033100

execution of other computer programs and provides scheduling, debugging, input/output control, accounting, compilation, storage assignment, data management and memory management, and communication control and related services. The processor and operating system defines computer platform for which application programs in other computer programming languages are written. The invention is not limited to any particular processor, operating system or programming language.

A memory system typically includes a computer readable and writeable nonvolatile recording medium in which signals are stored that define a program to be executed by the processor or information stored on the disk to be processed by the program. Typically, in operation, the processor causes data to be read from the nonvolatile recording medium into another memory that allows for faster access to the information by the processor than does the disk. This memory is typically a volatile, random access memory such as a dynamic random access memory (DRAM) or static memory (SRAM). The processor generally manipulates the data within the integrated circuit memory and may copy the data to the disk if processing is completed. A variety of mechanisms are known for managing data movement between the disk and the integrated circuit memory element, and the invention is not limited thereto. The invention is not limited to a particular memory system.

Such a system may be implemented in software or hardware or firmware, or any combination thereof. The various elements of this system, either individually or in combination, may be implemented as a computer program product including a computer-readable medium on which instructions are stored for access and execution by a processor. Various steps of the process may be performed by a computer processor executing instructions stored on a computer-readable medium to perform functions by operating on input and generating output.

Additionally, the computer system may be a multiprocessor computer system or may include multiple computers connected over a computer network. Various possible configurations of computers in a network permit access to the system by multiple users using multiple instances of the programs even if they are dispersed geographically. Each program or step shown in the figures and the substeps or subparts shown in the figures may correspond to separate modules of a computer program, or may be separate computer programs. Such modules may be operable on one or more separate computers or other devices. The data produced by these components may be stored in a memory system or transmitted between

computer systems or devices. The plurality of computers or devices may be interconnected by a communication network, such as a public switched telephone network or other circuit switched network, or a packet switched network such as an Internet protocol (IP) network. The network may be wired or wireless, and may be public or private.

5           A suitable platform for implementing software to provide such an authoring system includes a processor, operating system, a video capture device, a Creative Labs Sound Blaster or compatible sound card, CD-ROM drive, and 64 Megabytes of RAM minimum. For analog video capture, the video capture device may be the Osprey-100 PCI Video Capture Card or the Eskape MyCapture II USB Video Capture Device. The processor may be a 230  
10   megahertz Pentium II or Pentium III processor, or Intel equivalent processor with MMX Technology, such as the AMD-K6-III, or Celeron Processor with 128K cache, and may be used with an operating system such as the Windows98/98SE or Millennium operating systems. For digital video capture, the video capture device may be an IEEE 1394 Port (OHCI compliant or Sony iLink). The processor may be a 450 megahertz Pentium II or  
15   Pentium III processor, or Intel equivalent processor with MMX Technology, such as the AMD-K6-III, or Celeron processor with 128K cache.

          Given an authoring tool such as described above, the use of multiple authoring tools by multiple authors for publishing data to a public or private computer network for access by other users will now be described in connection with Figs. 12 and 13. In particular, an  
20   encoded presentation 1200 and associated data files 1202 may be transferred by a transfer tool 1204 to a streaming media server 1206 and a data server 1208. The transfer tool also may store preference data 1210 for the author with a profile manager 1212. The streaming media server 1206 and data server 1208 may be publicly accessible web servers accessible by web browsers 1214. Other kinds of distributed libraries of digital media, instead of a web  
25   server, also may be used to publish the presentation. If additional transfer tools 1216 are used by other authors, these transfer tools 1216 may transfer the streaming media to the same or a different streaming media data server 1206 as the other transfer tool 1204, but may have a separate data server 1218. Use of the same streaming media data server is possible where each transfer tool has access to the streaming media server 1206. Such access may be built  
30   into either the transfer tool or the authoring tool. The transfer tool and/or the authoring tool may be provided by the same entity or another entity related to the entity that owns or distributes the streaming media server 1206. The streaming media server may be

